Learning to Identify Non-Technical Losses with Optimum-Path Forest

Caio C. O. Ramos, André N. Souza
Department of Electrical Engineering
São Paulo State University
Bauru, Brazil
caioramos@gmail.com, andrejau@feb.unesp.br

João P. Papa, Alexandre X. Falcão
Institute of Computing
University of Campinas
Campinas, Brazil
papa.joaopaulo@gmail.com, afalcao@ic.unicamp.br

Abstract—In this work we have proposed an innovative and accurate solution for non-technical losses identification using the Optimum-Path Forest (OPF) classifier and its learning algorithm. Results in two datasets demonstrated that OPF outperformed the state of the art pattern recognition techniques and OPF with learning achieved better results for automatic non-technical losses identification than recently ones obtained in the literature.

Keywords-Non-Technical Losses, Optimum-Path Forest.

I. INTRODUCTION

Losses in electric power systems are constituted by the difference between the generated/bought energy and the billed ones, and can be divided into two distinct types: (i) technical and (ii) non-technical losses. The former are related with problems in the system through the physical characteristics of the equipment, that is, the technical losses are the energy lost in the transport, the transformation and the equipment of measurement, becoming a very high cost to the electric power companies [1]. The commercial losses or non-technical losses are those associated with the commercialization of the supplied energy to the user and refer to the delivered and not billed energy, propitiating a loss in the profits. They also are defined as the difference between the total losses and the technical losses, been strongly related to illegal connections in the distribution system [2].

Theft and problems with power meters, with the purpose to modify the registration of electric power, are the main causes of commercial losses in national and international electric power companies, evidencing the energy frauds. However, it is a hard task to calculate or measure the amount of the commercial losses, because in most part of the cases it is almost impossible to know where they occur [3]. The illegal connections of electric power are the reason of constant concern, both for the electric power companies and for regulatory agencies.

Nagi et al. [4] used Support Vector Machines – SVM [5] for detection of electricity theft, and Monedero et al. [6] proposed to use Artificial Neural Networks – ANN [7] together with statistical analysis for fraud detection in electrical consumption. A hybrid approach between Genetic Algorithms - GA and SVM was also applied for non-technical losses detection [8]. Despite the use of these artificial intelligence techniques have been increasing, some serious flaws of them need to be revisited [9].

Papa et al. [9] have presented a novel framework for graph-based classifiers that reduce the pattern recognition problem as an optimum path forest computation (OPF) in the feature space induced by a graph. These kinds of classifiers do not interpret the classification task as a hyperplanes optimization problem, but as a combinatorial optimum-path computation from some key samples (prototypes) to the remaining nodes. Each prototype becomes a root from its optimum-path tree and each node is classified according to its strongly connected prototype, that defines a discrete optimal partition (influence region) of the feature space. The OPF-based classifiers have some advantages with respect to the aforementioned classifiers: (i) one of them is free of parameters, (ii) they do not assume any shape/separability of the feature space and (iii) run training phase faster, which allows the development of real time applications for fraud detection in electricity systems.

Recently, we have addressed the non-technical losses identification by means of OPF [10]. However, there exist situations which limit the size of the training set, e.g., limited storage capacity devices. In such situations, we need to use small training sets, but it is also desirable to keep high accuracy levels in the test set. Papa et al. [9] have presented a learning algorithm for OPF classifier which can identify the most relevant samples from the training set using a third evaluating set. In such a way, the OPF classifier is designed using both training and evaluating sets and is validated over the test set. This work presents some additions regarding to the previous one [10], in which we propose to increase the performance of the commercial losses automatic identification by using a learning algorithm for OPF. Some comparisons among OPF, SVM with Radial Basis Function - RBF and linear kernels and ANN-MLP are also performed. The remainder of this paper is organized as follows. Section II describes the theory of Optimum-Path Forest and Section IV presents the dataset and recognition features used. Finally, experimental results and conclusions are stated in Sections V and VI, respectively.

II. OPTIMUM-PATH FOREST CLASSIFIER

Let Z_1, Z_2 and Z_3 be training, evaluation, and test sets with $|Z_1|$, $|Z_2|$ and $|Z_3|$ samples of a given dataset. As already explained, this division of the dataset is necessary to validate the classifier and evaluate its learning capacity from the errors. Z_1 is used to project the classifier and Z_3 kept unseen during the project. A pseudo-test on Z_2 is used to teach the classifier by randomly interchanging samples of Z_1 with misclassified samples of Z_2. After learning, it is expected an improvement in accuracy on Z_3.

Let $\lambda(s)$ be the function that assigns the correct label i, $i=1, 2,\ldots,c$, to any sample $s\in Z_1\cup Z_2\cup Z_3$, $S\subset Z_1$ be a set of prototypes from all classes, and v be an algorithm which extracts n features from any sample $s\in Z_1\cup Z_2\cup Z_3$ and returns a vector $\vec{v}(s)$. The distance $d(s,t)\geq 0$ between two samples, s and t, is the one between their corresponding feature vectors $\vec{v}(s)$ and $\vec{v}(t)$. One can use any distance function suitable for the extracted features. The most common is the Euclidean one.

Our problem consists of projecting a classifier which can predict the correct label $\lambda(s)$ of any sample $s\in Z_3$. Training consists of finding a special set $S^*\subset Z_1$ of prototypes and a discrete optimal partition of Z_1 in the feature space (i.e., an optimum-path forest rooted in S^*). The classification of a sample $s\in Z_3$ (or $s\in Z_2$) is done by evaluating the optimum paths incrementally, as though it were part of the forest, and assigning to it the label of the most strongly connected prototype.

A. Training

Let (Z_1, A) be a complete graph whose the nodes are the samples and any pair of samples defines an arc in $A=Z_1\times Z_1$. The arcs do not need to be stored and so the graph does not need to be explicitly represented. A path is a sequence of distinct samples $\pi_t=\langle s_1,s_2,\ldots,s_k\rangle$ with terminus at a sample t. A path is said trivial if $\pi_t=\langle t\rangle$. We assign to each path π_t a cost $f(\pi_t)$ given by a connectivity function f. A path π_t is said optimum if $f(\pi_t)\leq f(\tau_t)$ for any other path τ_t. We also denote by $\pi_s\cdot\langle s,t\rangle$ the concatenation of a path π_s and an arc(s,t).

We will address the connectivity function f_{max}.

$$f_{\max}\left(\langle s\rangle\right)=\begin{cases}0 \; if \;\; s\in S\\ +\infty \;\; otherwise\end{cases},$$
$$f_{\max}\left(\pi_s\cdot\langle s,t\rangle\right)=\max\left\{f_{\max}\left(\pi_s\right),d\left(s,t\right)\right\} \quad (1)$$

such that $f_{max}(\pi_s\cdot\langle s,t\rangle)$ computes the maximum distance between adjacent samples along the path $\pi_s\cdot\langle s,t\rangle$. The minimization of f_{max} assigns to every sample $t\in Z_1$ an optimum path $P^*(t)$ from the set $S\subset Z_1$ of prototypes, whose minimum cost $C(t)$ is

$$C\left(t\right)=\min_{\forall\pi_t\in(Z_1,A)}\left\{f_{\max}\left(\pi_t\right)\right\}. \quad (2)$$

The minimization of f_{max} is computed by Algorithm 1, called OPF algorithm, which is an extension of the general image foresting transform (IFT) algorithm [11] from the image domain to the feature space, here specialized for f_{max}. As explained in the Section I, this process assigns one optimum path from S to each training sample t in a non-decreasing order of minimum cost, such that the graph is partitioned into an optimum-path forest P (a function with no cycles which assigns to each $t\in Z_1/S$ its predecessor $P(t)$ in $P^*(t)$ or a marker *nil* when $t\in S$. The root $R(t)\in S$ of $P^*(t)$ can be obtained from $P(t)$ by following the predecessors backwards along the path, but its label is propagated during the algorithm by setting $L(t)= \lambda(R(t))$.

Algorithm 1 - OPF Algorithm

Input: A training set Z_1, λ-labeled prototypes $S\subset Z_1$ and the pair (v,d) for feature vector and distance computations.
Output: Optimum-path forest P, cost map C and label map L.
Auxiliary: Priority queue Q and cost variable cst.

```
For each s∈Z1/S, set C(s)←+∞.
For each s∈S, do
        C(s) ←0, P(s) ←nil, L(s) ←λ(s), and insert s in Q.
While Q is not empty, do
        Remove from Q a sample s such that C(s) is minimum.
        For each t∈Z1 such that t≠s and C(t)>C(s), do
                Compute cst←max{C(s), d(s,t)}.
                If cst<C(t), then
                        If C(t)≠+∞, then remove t from Q.
                        P(t) ←s, L(t) ←L(s) and C(t) ←cst.
                        Insert t in Q.
```

We say that S^* is an optimum set of prototypes when Algorithm 1 minimizes the classification errors in Z_1. S^* can be found by exploiting the theoretical relation between minimum-spanning tree (MST) and optimum-path tree for f_{max} [12].

By computing an MST in the complete graph (Z_1,A), we obtain a connected acyclic graph whose nodes are all samples of Z_1 and the arcs are undirected and weighted by the distances d between adjacent samples. The spanning tree is optimum in the sense that the sum of its arc weights is minimum as compared to any other spanning tree in the complete graph. In the MST, every pair of samples is connected by a single path which is optimum according to f_{max}. That is, the minimum-spanning tree contains one optimum-path tree for any selected root node.

The optimum prototypes are the closest elements of the MST with different labels in Z_1. By removing the arcs between different classes, their adjacent samples become prototypes in S^* and Algorithm 1 can compute an optimum-path forest with minimum classification errors in Z_1. Note that, a given class may be represented by multiple prototypes (i.e., optimum-path trees) and there must exist at least one prototype per class.

B. Classification

For any sample $t\in Z_3$, we consider all arcs connecting t with samples $s\in Z_1$, as though t were part of the training graph. Considering all possible paths from S^* to t, we find the optimum path $P^*(t)$ from S^* and label t with the class $\lambda(R(t))$ of its most strongly connected prototype $R(t)\in S^*$. This path can be identified incrementally, by evaluating the optimum cost $C(t)$ as

$$C\left(t\right)=\min_{\forall s\in Z_1}\left\{\max\left\{C\left(s\right),d\left(s,t\right)\right\}\right\}. \quad (3)$$

Let the node $s^*\in Z_1$ be the one that satisfies (3) (i.e., the predecessor $P(t)$ in the optimum path $P^*(t)$). Given that $L(s^*)=\lambda(R(t))$, the classification simply assigns $L(s^*)$ as the class of t. An error occurs when $L(s^*)\neq\lambda(t)$.

Similar procedure is applied for examples in the evaluation set Z_2. In this case, however, we would like to use misclassified samples of Z_2 to learn the distribution of the classes in the feature space and improve classification performance on Z_3.

III. Learning from Errors on the Evaluation Set

There are many situations that limit the size of Z_1: large datasets, limited computational resources, and high computational time as required by some approaches. Mainly in applications with large datasets, it would be interesting to select for Z_1 the most informative samples, such that the accuracy of the classifier is little affected by this size limitation. It is also important to show that a classifier can improve its performance along time of use, when we are able to teach it from its errors. This section presents a general learning algorithm which uses a third evaluation set Z_2 to improve the composition of samples in Z_1 without increasing its size.

From an initial choice of Z_1 and Z_2, the algorithm projects an instance I of a given classifier from Z_1 and evaluates it on Z_2. The misclassified samples of Z_2 are randomly selected and replaced by samples of Z_1 (under certain constraints). This procedure assumes that the most informative samples can be obtained from the errors. The new sets Z_1 and Z_2 are then used to repeat the process during a few iterations T. The instance of classifier with highest accuracy is selected along the iterations. The accuracy values $LA(I)$ obtained for each instance I form a *learning curve*, whose non-decreasing monotonic behavior indicates a positive learning rate for the classifier. Afterwards, by comparing the accuracies of the classifier on Z_3, before and after the learning process, we can evaluate its learning capacity from the errors.

The accuracies $LA(I)$, $I=1,2,\ldots,T$, are measured by taking into account that the classes may have different sizes in Z_2 (similar definition is applied for Z_3). If there are two classes, for example, with very different sizes and a classifier always assigns the label of the largest class, its accuracy will fall drastically due to the high error rate on the smallest class. Details about accuracy computation can be found in Papa et al. [9]. Algorithm 2 presents this learning procedure.

Algorithm 2 - OPF Learning Algorithm

Input: Training and evaluation sets, Z_1 and Z_2, labeled by λ, number T of iterations, and the pair (v,d) for feature vector and distance computations.
Output: Leaning curve LA and the OPF classifier with highest accuracy.
Auxiliary: Arrays FP and FN of sizes c for false positives and false negatives and list LM of misclassified samples.

1. Set $MaxAcc \leftarrow -1$.
2. For each iteration $I=1,2,\ldots,T$, do
3. $LM(s)= \leftarrow 0$
4. Train OPF/SVM/ANN-MLP with Z_1.
5. For each class $i=1,2,\ldots,c$, do
6. $FP(i) \leftarrow 0$ and $FN(i) \leftarrow 0$
7. For each sample $t \in Z_2$, do
8. Use the classifier obtained in Line 3 to classify t
9. with label $L(t)$.
10. If $L(t) \neq \lambda(t)$, then
11. $FP(L(t)) \leftarrow FP(L(t))+1$
12. $FN(\lambda(t)) \leftarrow FN(\lambda(t))+1$
13. $LM \leftarrow LM \cup t$.
14. Compute accuracy $LA(I)$ according Papa et al. [9].
15. If $LA(I)>MaxAcc$ then save the current instance
16. of the classifier and set $MaxAcc \leftarrow LA(I)$.
17. While $LM \neq 0$
18. $LM \leftarrow LM/t$
19. Replace t by a randomly selected sample of
20. the same class in Z1, under some constraints.

Line 4 is implemented by computing $S^* \subset Z_1$ as described in Section II-A and the predecessor map P, label map L and cost map C by Algorithm 1. The classification is done by setting $L(t) \leftarrow L(s^*)$, where $s^* \in Z_1$ is the sample that satisfies (3). The constraints in Lines 19-20 refer to keep the prototypes out of the sample interchanging process between Z_1 and Z_2. Lines 5-6 initialize the false positive and false negative arrays for accuracy computation. The classification of each sample is performed in Lines 7-13, updating the false positive and false negative arrays. Misclassified samples are stored in the list LM (Line 13). Line 14 computes the accuracy $LA(I)$ and Lines 15-16 save the best instance of classifier so far. The inner loop in Lines 17-20 changes the misclassified samples of Z_2 by randomly selected samples of Z_1, under the aforementioned constraints.

IV. Material and Methods

For the development of this work, we used two datasets obtained from a brazilian company of electric power, say that B_i and B_c datasets. The former is a dataset composed by 5190 industrial profiles, and the last one, i.e., B_c is composed by 8067 commercial profiles. Notice that both datasets were previously labeled by technicians of the aforementioned company. Each consumer profile is represented by four features, as follows:

- Contracted Demand: the value of demand to be continuously available by the energy company and shall be paid likewise whether the electric power is used or not by the consumer, in kilowatts (kW);
- Measured Demand or Maximum Demand (D_{max}): the maximum demand for active power, verified by measurement, at intervals of fifteen minutes during the billing period, in kilowatts (kW);
- Load Factor (LF): the ratio between the average demand ($D_{average}$) and maximum demand (D_{max}) of the consumer unit, recorded in the same time period;
- Installed Power (P_{inst}): the sum of the nominal power of electrical equipments installed and ready to operate at the consumer unit, in kilowatts (kW).

V. Experimental Results

We performed two series of experiments: in the former (Section V-A) we used 30% of the whole dataset for training and 50% for testing classifiers (the remaining 20% will be used in the next experiment), and in the last round (Section V-B) we executed the OPF with its learning algorithm (Section III) using 30% for training, 20% for the evaluation set and the remaining 50% for testing. For all experiments, we executed OPF, OPF with learning, SVM-RBF (SVM with RBF as kernel function), SVM-LINEAR (SVM with a linear kernel function) and ANN-MLP (ANN-MLP trained by backpropagation algorithm) 10 times with randomly generated training and test sets, to compute the mean accuracy and its standard deviation.

For SVM-RBF, we used the latest version of the LibSVM package [13] with Radial Basis Function (RBF) kernel, parameter optimization and the one-versus-one strategy for the multi-class problem. With respect to SVM-LINEAR, we used the LibLINEAR package [14] with default C=1 parameter. For OPF we used the LibOPF [15], which is a library for the design of optimum-path forest-based classifiers, and for ANN-MLP we used the Fast Artificial Neural Network Library (FANN) [16]. The network configuration is $i{:}h_1{:}h_2{:}o$, where i=4 (number of features), h_1=h_2=10 and o=2 (number of classes) are the number of neurons in the input, hidden and output layers, respectively. Note that we used here two hidden layers, i. e., h_1 and h_2. The ANN-MLP was trained with a backpropagation algorithm, and its architecture was empirically chosen.

A. Classifiers Evaluation

We evaluate here the OPF, SVM-RBF, SVM-LINEAR and ANN-MLP for non-technical losses detection using 30% for training and 50% for testing. Table I shows the mean accuracies and their standard deviations after 10 runnings with randomly generated training and test sets. We can see that OPF outperformed all classifiers in both B_c and B_i datasets with a large advantage.

TABLE I. MEAN ACCURACY AND STANDARD DEVIATIONS FOR OPF, SVM-RBF, SVM-LINEAR AND ANN-MLP

Classifier	*Accuracy B_i*	*Accuracy B_c*
OPF	83.31±1.92	84.48±1.48
SVM-RBF	74.21±3.03	75.87±2.51
SVM-LINEAR	48.23±2.81	53.46±3.10
ANN-MLP	52.01±5.95	73.33±3.48

B. OPF Learning Algorithm

In this section we evaluated the robustness of the OPF learning algorithm into learning the most relevant samples from the training set from an evaluating set. Table II presents the results. We can see that OPF with learning can provide better results than OPF without using the learning procedure.

TABLE II. MEAN ACCURACY AND STANDARD DEVIATIONS FOR OPF AND OPF WITH LEARNING

Classifier	*Accuracy B_i*	*Accuracy B_c*
OPF	83.31±1.92	84.48±1.48
OPF with learning	85.49±2.01	85.49±2.39

The OPF learning algorithm can finds the most relevant samples in a reduced training set by assuming that the misclassified ones in the evaluating set are the most informative samples. This affirmative states that a classifier can learn with its own errors and also can improve its performance over an unseen test set.

VI. CONCLUSIONS

In this work we have presented an innovative approach for improving the non-technical losses recognition rate by applying the OPF learning algorithm. Results in two datasets composed by commercial and industrial consumers demonstrated that OPF without the learning algorithm outperformed the state of the art pattern recognition techniques, i.e., Support Vector Machines and Artificial Neural Networks. Another round of experiments demonstrated that our proposed approach outperformed the recently results obtained by Ramos et al. [10].

REFERENCES

[1] A. H. Nizar, Z. Y. Dong, Y. Wang, “Power Utility Nontechnical Loss Analysis with Extreme Learning Machine Method”, *IEEE Transactions on Power Systems*, vol. 23, pp. 946-955, Aug., 2008.

[2] M. E. de Oliveira, D. F. A. Boson, A. Padilha-Feltrin, “A Statistical Analysis of Loss Factor to Determine the Energy Losses,” *IEEE/PES Transmission and Distribution Conference and Exposition: Latin America 2008*, pp. 1 – 6, Aug., 2008.

[3] A. H. Nizar, Z. Y. Dong, “Identification and Detection of Electricity Customer Behaviour Irregularities,” *IEEE/PES Power Systems Conference and Exposition 2009 – PES’ 09*, pp. 1-10, March, 2009.

[4] J. Nagi, A. M. Mohammad, K. S. Yap, S. K. Tiong, S. K. Ahmed, “Non-Technical Loss Analysis for Detection of Electricity Theft Using Support Vector Machines”, *IEEE 2nd International Power and Energy Conference 2008 - PECon 2008*, pp. 907 – 912, Dec., 2008.

[5] B. Boser, I. Guyon and V. Vapnik, “A Training Algorithm for Optimal Margin Classifiers”, *Proc. of the 5th Workshop on Computational Learning Theory*, ACM Press, pp. 144-152, 1992.

[6] I. Monederon, F. Biscarri, C. Léon, J. Biscarri, and R. Millán, “Midas: Detection of Non-Technical Losses in Electrical Consumption Using Neural Networks and Statistical Techniques”, *Proceedings of the International Conference on Computational Science and Applications*, vol. 3984, 2006.

[7] S. Haykin, “Neural Networks – A Comprehensive Foundation”, New York: Macmillan, 1994.

[8] J. Nagi, A. M. Mohammad, K. S. Yap, S. K. Tiong, S.K. Ahmed, “Detection of Abnormalities and Electricity Theft Using Genetic Support Vector Machines”, *IEEE Conference TENCON 2008*, pp. 1 – 6, Nov., 2008.

[9] J. P. Papa, A. X. Falcão, and C. T. N. Suzuki, “Supervised Pattern Classification Based on Optmum-Path Forrest”, *International Journal of Imaging Systems and Technology*, vol. 19, pp. 120–131, 2009.

[10] Caio C. O. Ramos, André N. de Souza, João P. Papa, Alexandre X. Falcão, “Fast Power Utility Non-Technical Losses Identification Through Optimum-Path Forest”, *Proceedings of 15th International Conference on Intelligent System Applications to Power Systems – ISAP*, Curitiba, 2009.

[11] A. X. Falcão, J. Stolfi, and R.A. Lotufo, “The Image Foresting Transform: Theory, Algorithms and Applications”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 19–29, 2004.

[12] P. Miranda, A. Rocha and F. Bergo, “Object Delineation by k-Connected Components”, *EURASIP Journal on Advances in Signal Process*, 2008.

[13] C. C. Chung and C. J. Lin, “LIBSVM: a Library for Support Vector Machine”. http://www.csie.ntu.edu.tw/cjlin/libsvm, March, 2009.

[14] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin, “LIBLINEAR: A Library for Large Linear Classification,” *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[15] João Paulo Papa, Celso Tetsuo Nagase Suzuki, and Alexandre Xavier Falcão, “LibOPF: A Library for the Design of Optimum-Path Forest Classifiers”, 2008. Software version 1.0 available at http://www.ic.unicamp.br/~afalcao/LibOPF

[16] S. Nissei, “Implementation of a Fast Artificial Neural Network Library (FANN)”, Department of Computer Science University of Copenhagen (DIKU), 2003. Software available at http://leenissen.dk/fann/.